AMENDMENTS TO THE CLAIMS

1. (Previously presented) A network adapter capable of receiving data from a network, said adapter arranged to receive said data at at least a lowest and a highest data rate; said adapter comprising:

sampling means arranged to sample said data and produce data samples;

an equalizer arranged to receive, and equalize, said data samples, and said equalizer capable of being trained to equalize said data, at at least, each of said lowest and highest data rates; and

training means capable of training said equalizer to equalize said data;

wherein;

said training means is initially arranged to train said equalizer to receive said data at said lowest data rate allowing data to be decoded and if upon decoding said adapter determines that said equalizer has been trained to equalize data at an incorrect rate to retrain said equalizer to equalize data at a correct rate.

2. (Previously presented) An adapter according to claim 1 that is arranged to receive one or more data sequences and is further arranged such that said equalizer is trained for each said data sequence received.

3. (Previously presented) An adapter according to claim 1 including a data buffer arranged to receive said data samples from said sampling means.

4. (Currently amended) An adapter according to claim 3 wherein said data buffer is arranged to store data at [[said]] a highest data frequency.

5. (Previously presented) An adapter according to claim 1 wherein said adapter also comprises a training sequence store buffer arranged to receive and store a training sequence held within said data.

6. (Previously presented) An adapter according to claim 5 wherein said training sequence store buffer is arranged to hold said training sequence at least until it has been determined that said equalizer has been trained to receive said data at said correct rate.

7. (Previously presented) An adapter according to claim 5 wherein said training sequence store buffer is arranged to receive data from said sampling means.

8. (Previously presented) An adapter according to claim 3 which also comprises a data down-sampling means arranged to output a selection of said data held in said data buffer to said equalizer.

9. (Previously presented) An adapter according to claim 5 which also comprises a data down-sampling means arranged to output a selection of said data held in said training sequence store buffer.

10. (Previously presented) An adapter according to claim 8 wherein said down-sampling means is arranged to output a selection of said data therein by predetermining a constant n, and outputting every nth data sample from said data buffer.

11. (Currently amended) A method of training an equalizer to equalize a data sequence, which said data sequence ~~may be~~ is at one of at least a highest and a lowest data rate, said method comprising:

obtaining at least one data sample by sampling said data sequence;

training said equalizer to receive said data samples at said lowest data rate to provide a trained equalizer;

decoding a portion of said data sequence with said trained equalizer to ascertain the correct data rate; and

retraining said equalizer if said equalizer has been incorrectly trained.

12. (Previously presented) A method according to claim 11 wherein said method comprises using training data within a preamble of said data sequence to train said equalizer.

13. (Previously presented) A method according to claim 11 wherein said received data sequence is buffered.

14. (Previously presented) A method according to claim 13 wherein said data is buffered at said highest data rate.

15. (Previously presented) A method according to claim 12 wherein training data is held in a training sequence store buffer.

16. (Previously presented) A method according to claim 11 wherein said method comprises sampling said data sequence at said highest data rate and subsequently using only a portion of the samples should said data sequence be at another data rate.

17. (Previously presented) A method according to claim 16 wherein said method determines a number n and uses only every nth data sample, in order to use only a portion of said samples.

18. (Previously presented) A method according to claim 11 wherein the highest and lowest data rates are part of the series xy, where x and y are integers.

19. (Previously presented) A method according to claim 18 wherein x is two.

20. (Previously presented) A method according to claim 11 wherein said equalizer is trained for each data sequence received.

21. (Previously presented) A method according to claim 11 wherein said method detects an end of sequence marker within said data sequence and once this is detected returns to a state of waiting to train said equalizer.

22. (Previously presented) A method according to claim 13 wherein said data sequence comprises a preamble, the end of which is received before buffering of said data samples is enabled.

23. (Previously presented) A computer readable medium having stored therein instructions for causing a processing unit to execute the method of claim 11.

24. (Currently amended) A computer program arranged to cause a data sequence to be received, which data sequence ~~may be~~ is at one of at least a highest and a lowest data rate, said program:

obtaining samples of said data sequence;

training an equalizer to equalize said samples to provide a trained equalizer;

decoding a portion of said data sequence using said trained equalizer; and

re-training said equalizer if the equalizer has been trained incorrectly at an incorrect data rate.

25. (Previously presented) An interface including an adapter according to claim 1.

26. (Previously presented) An interface according to claim 25 arranged to interface a computer or computer peripheral to a network.

27. (Previously presented) An interface according to claim 25 provided as any of the following: a PCI card, an ISA card, a USB peripheral, a Firewire peripheral, a PCMIA card, a MODEM riser card.

28. (Previously presented) A network adapter capable of receiving data from a network, said adapter arranged to receive said data at at least a lowest and a highest data rate; said adapter comprising:

a sampler arranged to sample said data to produce at least one data sample;

an equalizer arranged to receive, and equalize, said at least one data sample, and said equalizer capable of being trained to equalize said data, at at least, each of said lowest and highest data rates; and

a trainer capable of training said equalizer to equalize said at least one data sample; wherein

said trainer is arranged to initially train said equalizer to receive said data at said lowest data rate allowing said data to be decoded and if upon decoding said data said adapter determines that said equalizer has been trained to equalize said data at an incorrect rate to retain said equalizer to equalize said data at a correct rate.

29. (Previously presented) A network adapter capable of receiving data from a network, said adapter arranged to receive a plurality of data sequences each said data sequence containing said data, said adapter arranged to receive said data at at least a lowest and a highest data rate; said adapter comprising:

a sampler arranged to sample said data to produce at least one data sample;

an equalizer arranged to receive and equalize said at least one data sample, and said equalizer being capable of being trained to equalize said data at at least, each of said lowest and highest data rates and said equalizer being arranged to be trained for each data sequence received;

and a trainer capable of training said equalizer to equalize said at least one data sample wherein said trainer is arranged to initially train said equalizer to receive said data at said lowest data rate allowing a portion of said data to be decoded and if, upon decoding said portion of said data, said adapter determines that said equalizer has been trained to equalize said data at an incorrect rate said trainer is arranged to retrain said equalizer to equalize said data at a correct rate.

30. (Previously presented) An adapter according to claim 29 which comprises a training sequence store buffer arranged to receive an store a training sequence held within said data.

31. (Previously presented) An adapter according to claim 30 wherein said training sequence store buffer is arranged to hold said training sequence at least until it has been determined that said equalizer has been trained to receive said data at said correct rate.

32. (Previously presented) An adapter according to claim 31 wherein a data buffer is provided and arranged to buffer said data until is determined that said equalizer has been trained to receive said data at said correct rate.

33. (Previously presented) An adapter according to claim 29 wherein said portion of said data comprises a header containing said correct data rate at which said equalizer should be trained.

34. (Previously presented) A method of training an equalizer to equalize a plurality of data sequences, which said data sequences ~~may be~~ is at one of at least a highest and a lowest data rate, said method comprising:

obtaining at least one data sample by sampling said data sequence;

training said equalizer to receive said at least one data sample at said lowest data rate to provide a trained equalizer;

decoding a portion of said data sequence with said trained equalizer to ascertain the correct data rate;

retraining said equalizer if said equalizer has been incorrectly trained; and

restarting the training process once a data sequence has been completely received to allow the next data sequence to be received.

35. (Previously presented) A method according to claim 34 wherein said portion of said data sequence comprises a header containing said data rate at which said equalizer should be trained.